from the editors

Well, the (LISP BULLETIN) is back again. It was invented, a long time ago, by Daniel G. Bobrow, and a first issue was published in SIGPLAN, September 1969. The bulletin has exceptionnally deep roots and cannot be pulled free. So now here is the second issue (which explains the #2 on the cover). We think that the main purpose of the (LISP BULLETIN) is to make easier the communication between members of the LISP community.

We welcome contributions. The followins topics are particularly encouraged

- announcements for new books
- book and paper reviews
- small technical papers
- useful functions
- comments
- puzzles
- announcements for new LISP systems
- bars of silver
- precious jewelry

As you may notice we have not chansed our subscription rate (free), acknowledsins the frequency of publication. We will try to stick to a more resular schedule, so our subscription policy may be chansed in the future.

In the meantime, enjoy it.

(LISP PUZZLES

from P. GREUSSAY
This is the function SKE. What is she doins?

```
(DE ske (l r)
    (COND
       ((ATOM l) NIL)
       ((MEMQ l r) T)
       ((ske (CAR l) (CONS l r)) T)
       (T (ske (CDR l) (CONS l r)))))
```

from J. ALLEN
This is the function FOO. What is she doins?

```
(DE foo (l)
    (COND
       ((NULL l) NIL)
       ((NULL (CDR l)) l)
       (T (CONS (CAR (foo (CDR l)))
                (foo (CONS (CAR l)
                           (foo (CDR (foo (CDR l))))))))))
```

from H. SAMET
This is the function BAR. What is she doins?

```
(DE bar (x y)
   (IF (< x 2) (ADD1 y)
       (bar (SUB1 x) (bar (- x 2) y))))
```

from D. GOOSSENS
This is the function MOBY. What is she doins?

```
(DE moby (l)
    (IF (NULL (CDR l))
        (CAR l)
        (moby (CDDR (APPEND l [(CAR L)])))))
```

from H. BOLEY
Write a LISP function NOTHING without parameter such that the call (NOTHING) does 'nothins', i.e. no value [includins NIL] is returned and no effect on the further interaction with LISP is noticed. So in LISP 1.6 for.example NOTHING should enable the followins interaction :

```
* (SETQ X 2)
2
* (NOTHING)
* X
2
* ...  further normal LISP 1.6 interaction ...
```

SEND MORE PUZZLES)

(LISP BOOKS

ALLEN J., The Anatomy of LISP, McGraw Hill, New York 1977

BERKELEY E.C. & BOBROW D.G., (ed), The Programming Language LISP : Its Operation and Applications, Information International Inc., The M.I.T. Press, Cambridge, Mass., 1964

FRIEDMAN D.P., The little lisper, Science Research Associates Inc., 1974

MAURER W.D., The Programmer's Introduction to LISP, Mac Donald / American Elsevier Computer Monographs, 1972

Mc CARTHY J. & TALCOTT C., LISP Programming and Proving, Stanford University, march 1978

Mc CARTHY J. et al, LISP 1.5 Programmer's Manual, The M.I.T. Press, Cambridge, Mass., 1962

NAKANISHI M., Initiation to LISP, Modern Science Co., Tokyo, Japan, 1977

PETER R., Rekursive Funktionen in der Komputer-Theorie, Akademiai Kiado, Budapest, 1976

RIBBENS D., Programmation non-numerique LISP 1.5, Dunod, Paris, 1969

SIKLOSSY L., Let's Talk LISP, Prentice Hall, Englewood Cliff, New Jersey, 1976

STOYAN H., LISP Programmier-Handbuch, Akademie Verlag, Berlin, DDR, 1978

WEISSMAN C., LISP 1.5 Primer, Dickenson Publishing Company Inc., Belmont, California, 1967

WINSTON P.H., Artificial Intelligence, Addison Wesley, 1977

WRITE MORE LISP BOOKS)

Book review

JOHN ALLEN, The Anatomy of LISP,
McGRAW HILL, New York, 1977.

At last there exists a textbook for the computer scientist that gives a self-contained and in-depth treatment of "LISP" in its broadest sense. ALLEN's exposition of LISP serves as a manifestation for fundamental ideas such as: structured-programing and step-wise refinement, abstract programs and abstract data, data-driven programing, proving properties of algorithms, programing language semantics, translator implementation etc. This text will enable the student to regain the perspective he possibly lost in studying isolated topics or never gained in other so-called "introductions". Augmented by projects, "The Anatomy of LISP" is the best introduction to computer science I have seen.

After introducing symbolic expressions and a few LISP applications (mainly in algebraic manipulation), a series of languages together with their evaluators is built. ALLEN starts out with the language of polynomials, and a function to compute their value, and then proceeds step by step adding all the features usually found in programing languages, until advanced constructs like Label, closures and non-recursive control-structures. Each time the language is altered, so is its evaluator. The evaluators are rather concise and leave implementation issues for refinement to a special chapter. The semantics of a newly introduced construct is precisely defined by the new interpreter/evaluator. ALLEN's presentation strategy breaks the habit by which we acquire natural language constructs, namely through perceiving their usage in context. Breaking this habit is a healthy first step towards understanding the creation of new or more general language constructs. The working of these evaluators is clearly visualized using a representation of environments also known from WEIZENBAUM's explanation of FUNARGs [WEI].

The chapter on implementation covers solutions to the common problems and pitfalls encountered when implementing high-level programing languages. The discussion of binding is very profound and can be recommended as a first reading before studying more specialized papers such as [B&W], [STE] etc.. Most of this chapter discusses topics also relevant to other areas of software engineering: symbol-tables, table-searching, storage management and syntax-directed Input/Output.

The book ends with a series of interesting projects for students. This collection could be augmented by more typical AI-problems which would provide motivation for applying the idea of language and evaluator design as a means for problem-solving.

[B&W] Bobrow, D. and Wegbreit, B. A model and stack implementation of multiple environments, CACM, 16, 591-603.

[STE] Steele, G.L. Jr., Macaroni is better than spaghetti, Proc. of the Symposium on Artificial Intelligence and Programing Languages, ACM, 1977, 60-66.

[WEI] Weizenbaum, J. The Funarg problem explained, Intern. Seminar on Adv. Progr. Syst., Jerusalem, 1968.

(reviewed by: Joachim Laubsch)

(CURRENT LISP THESIS

CARTWRIGHT R. Jr, Formal Semantics of LISP with Applications to Program Correctness, Stanford University, Artificial Intellisence Laboratory, AIM-257, January 1975

GREUSSAY P., Contribution a la definition interpretative et a l'implementation des lambda-lansases, These d'Etat, Universite Paris 7, Novembre 1977, Rapport L.I.T.P #7

LECOUFFE P., Etude et Definition d'une Machine Lansase LISP, These de 3eme Cycle, Universite de Lille, Decembre 1977

LUX A., Etude d'un modele abstrait pour une machine LISP et de son implementation, These de 3eme Cycle, Universite de Grenoble, Mars 1975

NEWEY M. C., Formal Semantics of LISP with Applications to Program Correctness, Stanford University, Artificial Intellisence Laboratory, AIM-257, January 1975

SAMET H., Automatically Proving the Correctness of Translations Involving Optimized Code, Stanford University, Artificial Intellisence Laboratory, AIM-259, May 1975

TERASHIMA M., Algorithms Used in an Implementation of HLISP, Information Science Laboratory, Faculty of Science, University of Tokyo, January 1975

WERTZ H., Un Systeme de Comprehension, d' Amelioration et de Correction de Programmes Incorrects, These de 3eme Cycle, Universite Paris 6, Juillet 1978

WRITE MORE LISP THESIS)

(ANNOUNCEMENTS FOR LISP SYSTEMS

R.R. JOHNSON
Dept. of Computer Science
University of Kentucky
Lexington, Kentucky 40506

We are putting together a LISP-system on our Varian 74 mini-computer. Our primary interest is to use the micro-programming capabilities of the machine to implement some of the most used parts of the LISP-system.

S.S. MUCHNICK
Dept of Computer Science
The University of Kansas
Lawrence, Kansas 66045

An implementation effort for LISP 1.5 is now underway on an Interdata 85. The interpreter is being written in PL/85, a locally developed structured assembly language. Though the implementation is being done on a model 85 it is compatible with other Interdata 16-bit processors. The basic interpreter, storage management, and input/output packages are complete and we are currently extending our collection of SUBRs. We intend to implement the modified interpreter described in P. Greussay, Iterative Interpretation of Tail-Recursive LISP Procedures, as well.

Announcement

A New LISP System for IBM System 360/370

Over the last eighteen months we have developed in Cambridge, England a new IBM specific LISP system, with a number of advanced features. These include a built-in pretty printer for echoing of the input, which is also available to the user to print list structures. It has extensive and comprehensive tracing, backtrace and error recovery facilities, and is designed to be safe. Errors such as taking the car of an atom are trapped immediately, a feature which is also available in compiled functions.

The system has extensive numerical features, such as arbitrary precision integers and rational numbers, as well as double precision floating point numbers and finite field functions, (arithmetic modoulo a prime).

The compiler is a version of the Griss and Hearn portable LISP compiler, which produces compiled functions as a normal LISP object held in the heap. To manage this there is an efficient compacting garbage collector.

The system is value cell LISP, and is implemented in a high level language, (BCPL). It runs interpretively in 150Kbytes, and with the compiler in about 220Kbytes. Measurements have shown it to be efficient, (a little slower than Stanford LISP/360 when interpreted, faster when compiled) and it has been running successfully under OS in Cambridge for nine months. For further information contact one of:

Dr. John Fitch and Dr. Arthur Norman,
The Computer Laboratory,
University of Cambridge,
Corn Exchange Street,
Cambridge,
England.

14.1.77 JPF, ACN.

LISP for Interdata M85, 7/32

G. Persch, Gg. Winterstein

At the University of Kaiserslautern we have designed and implemented a LISP-system ID-LISP which runs on Interdata M85 and 7/32 minicomputers.

The Interdata M85 machine is a byte-oriented minicomputer with 64 K-byte working storage. It has 16 registers à 16 bit. The average time for an executable instruction is about 1 µs. The machine configuration at Kaiserslautern is for interactive use only and consists of a Hewlett Packard 264oA terminal and a Diabolo 162o for I/O, 4 disks and an Intertape Cassette unit. ID-LISP was designed in a way that it can run under the primitive operation system BOSS.

ID-LISP contains LISP 1.5 as a subset and has special features from MACLISP and INTERLISP as well. ID-LISP has its own EDITOR, PRETTYPRINT, file-handling (LOAD-SAVE) and various TRACE- and BREAK-functions. In the current version there is no paging.

The storage allocation is as follows:

hex.address	contents	size
oooo - 17oo	operating system BOSS	5.75 K-byte
17oo - 2EOO	programm-code	5.75 K-byte
2Eoo - 38oo	I/O-Buffer Hash-tables	2.5 K-byte
38oo - E8oo	11 K-LISP-cells	44. K-byte
E8oo - FFFF	stack	6. K-byte
		64. K-byte

4 bytes form a LISP-cell. Both pointers are absolute addresses. All parts of a symbol are represented in LISP-cells. For every symbol we have

(<list of bindings><function-definition><Pname><property>...)

Symbols are respresented only once. They are identified through a hash-table with Add-the-Hash-Rehash. Numbers are also stored within the LISP-storage. Small numbers (14 bit) are directly represented in the pointer. The garbage-collection-algorithm is a modification of P. Deutsch's pointer-chasing-algorithm.
At the time being the system is only available on cassettes. There is also a manual available which is written in German. For more details and further informations contact the autors at

Universität Kaiserslautern
Fachbereich Informatik
Pfaffenbergstr. 95

D-675o Kaiserslautern, FRG

The use of LISP at computer centers in Western Germany

(A summary of G.GÖRZ "Die Verwendung von LISP an wissenschaftlichen Rechenzentren in der BRD", IAB Nr 63, Universität Erlangen-Nürnberg, Rechenzentrum, Dez. 76).

LISP-systems as used on various computer systems

Computer	LISP-System	Installation
Burroughs B67ØØ	LISP B 67ØØ	Inf., Karlsruhe
CDC : CD 33ØØ	LISP 1.5	U Erlangen
	LISP FINT	U Erlangen
	LISP F 1.1	U Gießen
		U Tübingen
CDC : CYBER bzw. 6ØØØ	LISP 1.5.6	TU Berlin
		RRZN Hamburg
	LISP 1.5.9	U Köln
	UTLISP 4.Ø	TU Berlin
		U Stuttgart
CGK : TR44Ø	B&LISP (1973)	GMD Darmstadt
		U ERlangen
		Inf. München
		U Saarbrücken
		U Tübingen
		U Ulm
	B&LISP (1976)	GRZ, Berlin
	LISPSYSTEM	Inf. München
	LISP 44Ø	IMMD Erlangen
	MACLISP	U Bielefeld
		U Bochum
		U Erlangen
		U Kaiserslautern
		Inf. Stuttgart
DEC : SYSTEM 1Ø	LISP 1.6 (II)	Inf. Hamburg
	LISP 1.6 (28.7.)	U Kiel
IBM 36Ø, 37Ø	LISP/36Ø (Stanford)	U Bielefeld
		U Bonn
		MPI, Garching
		KfA, Jülich

	LISP/36Ø (Stanford/36, Utah-Mod.)	GfK, Karlsruhe U Münster
	LISP 1.5/CMS (Grenoble)	Inf.,TU Berlin
	LISP (bits)	GMD, St. Augustin
	LISP FINT	U Bonn
	LISP F2	U Heidelberg
Interdata M85, 7/32	ID-LISP	U Kaiserslautern
Philips Electrologica X8	LISP-X8	U Kiel
	LISP	U Regensburg
Siemens 4ØØ4 (BS1ØØØ)	LISP F2	IdS, Mannheim
Siemens 4ØØ4/151 (BS2ØØØ)	INTERLISP	IdS Mannheim
Univac 11Ø8	11ØØ LISP	U Freiburg GWD, Göttingen U Karlaruhe

Applications

theorem-proving
GRZ-Berlin, U. Kaiserslautern
program-manipulation
GMD-St. Augustin, Informatik-Karlsruhe
natural-language processing
Inf.-Hamburg, U-Heidelberg, U-Köln,
Inst. f. deutsche Sprache (IdS), Inf. Stuttgart
cognitive psychology
Phychol. Inst. Uni Hamburg, Inf. Stuttgart
others: Computer-aided instruction, REDUCE, nuclear physics.

The report contains a brief description of each LISP-system and contact addresses for further reference. To obtain it write to G.Görz, RZ d Uni. Erlangen-Nurnberg, INFRA, Martenstr. 1, 8520 Erlangen.

LISP-Reference Manuals (in german)

(1) INTERLISP - Programmierhandbuch
Institut für deutsche Sprache
Abt. Linguistische DV,
Postfach 5409, D6800 MANNHEIM 1

(2) MACLISP - Reference Manual
J. Laubsch, Inst. f. Informatik
Azenbergstr. 12, D7000 Stuttgart 1

Jerome CHAILLOUX
Universite de Paris VIII - Vincennes
Route de la Tourelle
75571 Paris Cedex 12 (France)

A VLISP System for 8-bit words micro-computers.

A new version of the VLISP system, VLISP 8, has been implemented on the 8-bit micro-computers
- Intel 8080
- Zilog 80

VLISP 8 is available for the followins systems :
- MDS ISIS 1 and 2 (a 8080 based system), with the minimal confisuration :
  - 32k RAM
  - a teletype
  - one floppy disk
- MOSTEK DDT80 (a Z80 based system), with the minimal confisuration :
  - 16k RAM
  - 8k REPROM
  - a teletype

The interpreter occupies 8k bytes. Special attention has been paid to speed up the evaluation of forms. In particular, the interpreter does not perform any internal CONSes.
Naturally, this system owns all the new features recently introduced into the other VLISP systems, like ESCAPE LESCAPE SELF ... , as well as the iterative interpretation of tail-recursive functions calls.
This system is already used to introduce children to program in VLISP and a LOGO-like lansuase.

SEND MORE ANNOUNCEMENTS FOR LISP SYSTEMS)

(CURRENT LISP MANUALS

CHAILLOUX J., VLISP-10 Manuel de Reference, Dept. Informatique, Universite Paris-8 Vincennes, RT 17-76, 1976

DURIEUX J.-L., TLISP IRIS 80, Universite Paul Sabatier, Toulouse, 1977

GREUSSAY P., LISP T-1600 Manuel de Reference Provisoire, Universite Paris-8 Vincennes, RT 10-75, 1975

HARALDSON A., LISP-details INTERLISP / 360-370, Uppsala University, 1975

LAUBSCH J.H., MACLISP Manual, CUU-Memo-3, Universitaet Stuttgart, 1976

LUX A., LISP IRIS-80 Manuel d'Utilisation, february 1978, Universite de Grenoble, LA CNRS no 7

MOON D.A., MACLISP Reference Manual, M.I.T. Project Mac, Cambridge, Mass., 1974

QUAM L.M. & DIFFIE W., Stanford LISP 1.6 Manual, Stanford AI Project Operating Note 28.7, Computer Science Dept., Stanford University, 1972

TEITELMAN W., INTERLISP Reference Manual, XEROX Palo Alto Research Center, Palo Alto, Ca., 1974

SEND MORE CURRENT LISP MANUALS)

INTERLISP - Programmierhandbuch

In deutscher Sprache liegt auf 340 Seiten

- ein bewährtes Lehr- und Ausbildungswerk vor, das besonders zum Selbststudium geeignet ist und das

- ein Nachschlagewerk darstellt, das für alle, die INTERLISP benutzen, unentbehrlich ist.

Inhalt:

Kontaktadresse: Institut für deutsche Sprache
Abteilung Linguistische Datenverarbeitung
Forschung und Entwicklung
Friedrich-Karl-Str. 12, Postf. 5409

6800 Mannheim 1
B R D